REMARKS

The disclosure is objected to because of certain informalities. The specification is amended at page 6, lines 8-10 as suggested by the Examiner. No new matter is introduced by the amendments to the specification.

Claim 1 is rejected under 35 U.S.C. 112, second paragraph, for informalities stated in the Office Action at page 2, section 4 to page 3, section 8. The claims are amended such that the issues related to antecedent basis are addressed. Reconsideration of the rejections is requested.

Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Rotenberg, *et al.* ("Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching," Proc. of the 29th International Symposium on Microarchitecture, Dec. 1996 - hereinafter "Rotenberg"), in view of Nair (U.S. Patent No. 6,304,962), and further in view of Gabzdyl, *et al.* (U.S. Patent No. 6,145,076 - hereinafter "Gabzdyl"). In view of the amendments to claim 1 and the following remarks, it is believed that the claim is allowable over the cited references. Accordingly, reconsideration of the rejection of claim 1 is respectfully requested.

The applicants' invention is directed a method of executing instructions using an address trace cache in a branch prediction environment. As noted in the applicants' Amendment After Final Rejection at page 7, second paragraph, dated September 3, 2004, in the present invention, when a routine is to be executed repeatedly, the addresses of the instructions are not stored in order in the address trace cache. In this manner, with regard to this type of routine of the present invention (referred to as a "second routine"), the routine start address, the routine end address, the current number of executed iterations, and the total number of iterations are stored in the address trace cache.

The claims are amended to clarify certain features of the invention. Specifically, the claims are amended to clarify that the routine start address is stored in a first storing area of the address trace cache, the routine end address is stored in a second storing area of the address trace cache, the current iteration count is stored in a third storing area of the address trace cache, and a total loop iteration count is stored in a fourth storing area of the address trace cache. In addition,

the claims are amended to clarify that the address trace cache provides a first storing area, second storing area, third storing area, and fourth storing area for each routine composed of a group of instructions to be repeatedly executed. In addition, the claims are amended to clarify that when the current iteration count and the total loop iteration count are identical, the second routine is finished, and a next routine composed of a next group of instructions to be repeatedly executed is initiated by addressing a start address in a next first storage area of the next routine. It is believed that these claim amendments clarify the patentable distinctions between the applicants' invention and the cited references.

It is therefore submitted that Rotenberg fails to teach or suggest certain elements of the present invention as claimed in claim 1. In particular, it is submitted that Rotenberg fails to teach or suggest storing a routine start address in a first storing area of a address trace cache, storing a routine end address in a second storing area of the address trace cache, storing a current iteration count of the second routine in a third storing area of the address trace cache, the current iteration count representing a current number of executed iterations of the second routine, and storing a total loop iteration count in a fourth storing area of the address trace cache, as claimed. While Rotenberg teaches a method of executing instructions using a trace cache, as noted in the Office Action at page 4, section 11(b), Rotenberg fails to teach that if a second routine is composed of a second group of instructions is to be repeatedly executed, the trace cache stores a routine start address, routine end address, current iteration count, and total loop iteration count.

In addition, since Rotenberg fails to teach or suggest the address trace cache storing the abovementioned second routine information in a first storing area, second storing area, third storing area, and fourth storing area, respectively, it follows that Rotenberg fails to teach or suggest the address trace cache providing a first storing area, second storing area, third storing area, and fourth storing area for each routine composed of a group of instructions to be repeatedly executed, as claimed.

In addition, it is submitted that Rotenberg fails to teach or suggest that when the current iteration count and the total loop iteration count representing the total number of iterations are identical, the second routine is finished, and a next routine composed of a next group of

instructions to be repeatedly executed is initiated by addressing a start address in a next first storage area of the next routine, as claimed. There is no mention in Rotenberg of either a current iteration count or a total loop iteration count. Moreover, since Rotenberg fails to teach the current iteration count and the total loop iteration count being identical, it follows that Rotenberg fails to teach or suggest that following the current iteration count and the total loop iteration count being identical, a next routine is initiated by addressing a start address in a next first storage area of the next routine, as claimed.

Nair is cited in the Office Action at page 4, lines 6-7, as teaching a Superblock Target Buffer (STB), comprising a cache for storing instruction addresses. However, it is submitted that Nair, like Rotenberg, fails to teach or suggest storing a routine start address in a first storing area of an address trace cache, storing a routine end address in a second storing area of the address trace cache, storing a current iteration count of the routine in a third storing area of the address trace cache, and storing a total loop iteration count in a fourth storing area of the address trace cache, as claimed. Specifically, there is no mention in Nair of a routine start address, a routine end address, a current iteration count, or a total loop iteration count, nor is there any mention of partitioning the cache into four separate storing areas for storing the routine start address, routine end address, current iteration count, and total loop iteration count, respectively.

In addition, it is submitted that Nair, like Rotenberg, fails to teach or suggest the address trace cache providing a first storing area, second storing area, third storing area, and fourth storing area for each routine composed of a group of instructions to be repeatedly executed, as claimed, for similar reasons as described above.

In addition, it is submitted that Nair, like Rotenberg, fails to teach or suggest that when the current iteration count and the total loop iteration count representing the total number of iterations are identical, the second routine is finished, and a next routine comprised of a next group of instructions to be repeatedly executed is initiated by addressing a start address in a next first storage area of the next routine, as claimed.

Gabzdyl is directed to a data processing circuit arranged to execute program instructions defining nested loops, wherein a loop includes a start address, an end address, and a number of

loop iterations (see Gabzdyl Abstract). However, it is submitted that Gabzdyl, like Rotenberg and Nair, fails to teach or suggest four distinct storing areas. Thus, it follows that Gabzdyl likewise fails to teach or suggest storing a routine start address in a first storing area of the address trace cache, storing a routine end address in a second storing area of the address trace cache, storing a current iteration count of the routine representing a current number of executed iterations of the routine in a third storing area of the address trace cache, and storing a total loop iteration count in a fourth storing area of the address trace cache, as claimed. Instead, Gabzdyl discloses three registers that are used for storing the start address, end address, and number of loop iterations for each nested loop function: a start address register 502 for storing the start address, an end address register 503 for storing the end address, and a counter 504 for storing the number of loop iterations (see Gabzdyl, Figure 5, and column 4, lines 3-6). There is no mention in Gabzdyl of storing the start address, end address, or number of loop iterations in respective storing areas of the address trace cache. In fact, there is no mention in Gabzdyl of an address trace cache.

In addition, it is submitted that Gabzdyl, like Rotenberg and Nair, fails to suggest the address trace cache providing a first storing area, second storing area, third storing area, and fourth storing area for each routine composed of a group of instructions to be repeatedly executed. While Gabzdyl discloses a start address register 502, an end address register 503, and a counter 504, there is no mention in Gabzdyl of providing different registers to serve as storing areas for the start address, end address, and number of loop iterations for each nested loop. Moreover, since Gabzdyl fails to teach an address trace cache, there is no need in Gabzdyl to provide storing areas for each routine of repeated instructions.

In addition, it is submitted that Gabzdyl, like Rotenberg and Nair, fails to teach that when the current iteration count and the total loop iteration count representing the total number of iterations are identical, the second routine is finished, and a next routine comprised of a next group of instructions to be repeatedly executed is initiated by addressing a start address in a next first storage area of the next routine, as claimed. Instead, as noted in the Office Action at page 5, section 11 (c), Gabzdyl decrements the count value held in counter 504 until the count is zero

(see Gabzdyl, column 5, lines 17-23). However, since Gabzdyl fails to teach or suggest different storage areas for each routine having repeatedly executed instructions, it follows that Gabzdyl fails to teach or suggest addressing a start address in a next first storage area of the next routine.

It is therefore submitted that neither Rotenberg, Nair, nor Gabzdyl teaches or suggests elements of claim 1 set forth above. Specifically, none of the references teaches or suggests the step of storing a routine start address in a first storing area of the address trace cache, storing a routine end address in a second storing area of the address trace cache, storing a current iteration count of the routine representing a current number of executed iterations of the routine in a third storing area of the address trace cache, and storing a total loop iteration count in a fourth storing area of the address trace cache, as claimed. In addition, none of the references teaches or suggests the address trace cache providing a first storing area, second storing area, third storing area, and fourth storing area for each routine composed of a group of instructions to be repeatedly executed, as claimed. In addition, none of the references teaches or suggests incrementing the current iteration count for each executed iteration of the second routine, wherein when the current iteration count and the total loop iteration count representing the total number of iterations are identical, the second routine is finished, and a next routine composed of a next group of instructions to be repeatedly executed is initiated by addressing a start address in a next first storage area of the next routine, as claimed.

Since the Rotenberg, Nair, and Gabzdyl references fail to teach or suggest these claimed features, there is no way to combine the references to obtain teaching or suggestion of the claimed features, and therefore, there is no combination of the references that teaches or suggests the invention as set forth in the amended claims.

Since Rotenberg, Nair, and Gabzdyl references, taken alone or in combination, fail to teach or suggest the present invention set forth in the amended claims, claim 1 is believed to be allowable over the cited references. Accordingly, reconsideration of the rejection of claim 1 under 35 U.S.C. 103(a) based on Rotenberg, Nair, and Gabzdyl is respectfully requested.

In view of the amendments to the specification and the claims and the foregoing remarks, it is believed that all claims pending in the application are in condition for allowance, and such allowance is respectfully solicited. If a telephone conference will expedite prosecution of the application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

Date: 5/20/05

Mills & Onello, LLP
Eleven Beacon Street, Suite 605
Boston, MA 02108
Telephone: (617) 994-4900
Facsimile: (617) 742-7774
J:\SAM\0169\amendb\amendmentb.wpd

Steven M. Mills
Registration Number 36,610
Attorney for Applicants